

## Aplicația II.3

### NOȚIUNI ȘI PROCEDURI AVANSATE PENTRU PROGRAMARE ÎN LABVIEW; REALIZAREA DE APLICAȚII COMPLEXE

Dacă în aplicațiile precedente s-a lucrat cu date sub formă de scalari, în cadrul acestei lucrări se vor prezenta unele noțiuni și proceduri privind modalități de grupare a datelor sub formă de șiruri și tabele. De asemenea, se vor face referiri la principalele structuri de programare care introduc facilități importante în ceea ce privește fluxul de date în diagrama bloc. Utilizarea acestor noțiuni și proceduri, pentru stocare, transfer și prelucrare de date, va fi ilustrată prin exemple din domeniul calculelor de o complexitate mai mare, cum sunt cele statistice.

#### 3.1 Gruparea datelor în LabVIEW

În raport de complexitate și de necesitățile de prelucrare, în mediul **LabVIEW** se pot aplica trei modalități de grupare a datelor, denumite în limba engleză **String**, **Array** și **Cluster**. În continuare, se vor folosi denumirile în engleză, întrucât, pe de o parte traducerile în română implică utilizarea mai multor cuvinte, iar pe de altă parte sunt mai ușor de recunoscut titlurile din palete, meniuri sau comenzi.

##### **String-uri**

Prin **String**, în traducere curentă **șir**, se înțelege o succesiune de caractere ASCII care, de regulă, reprezintă un text în care pot fi și caractere care nu se afișează. **String**-urile se folosesc pentru crearea de mesaje simple de text, pentru transferul datelor numerice către anumite instrumente, pentru stocarea datelor pe disc, pentru inițierea unor instrucțiuni etc.

Pe panoul frontal, **String**-urile apar sub formă de tabele, de casete pentru introducerea de texte, sau de etichete. Editarea și manipularea **String**-urilor se face în panoul frontal cu subpaleta **String&Path** din paleta **Controls**, procedând la fel ca pentru controalele și indicatoarele numerice, folosind **Operate Value** și **Edit Text** din paleta **Tools**. Executând click dreapta deasupra unui control sau indicator **String** se deschide un meniu care conține mai multe opțiuni de afișare, descrise în tabelul următor:

<i>Opțiunea din listă</i>	<i>Descrierea modului de afișare</i>
<b>Normal Display</b>	Afișează caractere care pot fi tipărite cu fontul atribuit controlului sau indicatorului
<b>'\ ' Codes Display</b>	Afișează codificat cu diagonală la stânga ( \ ) toate caracterele

	netipăribile
<b>Password Display</b>	Afișează câte un asterisc (*) pentru fiecare caracter, inclusiv spațiile
<b>Hex Display</b>	Afișează valoarea în ASCII a fiecărui caracter în hexa în locul caracterului în sine

În paleta **Controls** se află subpaleta **List&Table**, prin accesarea căreia se poate insera un tabel în panoul frontal. Fiecare locație care se află într-o linie sau într-o coloană din tabel conține un **string**, astfel că tabelul afișează o matrice 2D de **String**-uri.

În diagrama bloc pentru editarea și operarea cu **string**-uri se accesează diferitele opțiuni care sunt afișate în subpaleta **String** din paleta **Functions**, așa după cum se arată în fig.3.1

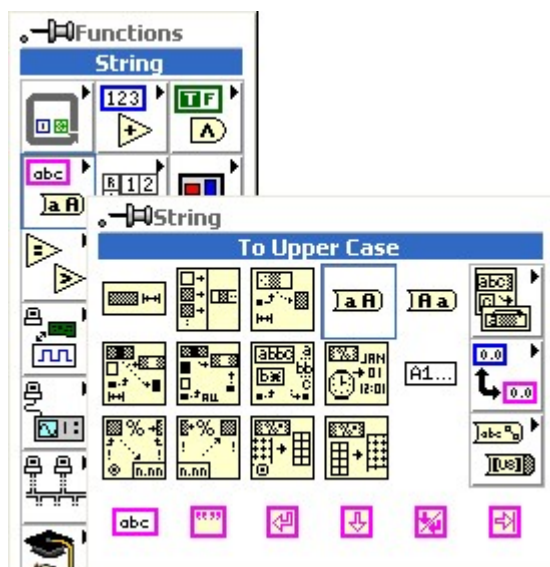
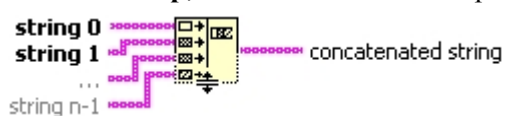


Fig. 3.1 Subpaleta **String** din paleta **Functions**

Icoanele conținute în subpaleta **String** simbolizează funcții cu care se pot executa diverse operații asupra șirurilor de caractere alfanumerice, atunci când o astfel de icoană este trasă în diagrama bloc. Semnificațiile acestor funcții și detalii asupra operațiilor pe care le efectuează se pot obține selectând o anumită icoană și apoi utilizând **Context Help**. De exemplu, dacă se selectează cea de a doua icoană de pe prima linie a subpaletei, aceasta corespunde funcției **Concatenate Strings**. Accesând **Context Help**, în fereastra acestuia apare imaginea din fig.3.2 și explicații privind



operațiile pe care le realizează această funcție.

Funcția **Concatenate Strings** posedă un număr de intrări ce poate fi modificat de către utilizator prin

Fig.3.2. Funcția **Concatenate Strings**

dimensionarea simbolului funcției. La ieșirea sa, funcția generează o valoare alfanumerică obținută prin concatenarea textelor de la intrările sale, în ordinea de sus în jos.

### Array-uri

Spre deosebire de elementele scalare, care reprezintă o singură valoare, de un anumit tip, elementele vectoriale/matriceale pot cuprinde, într-o manieră ordonată, mai multe valori de același tip și se reprezintă sub forma unui tablou (tabel), denumit în limba engleză **Array**. Un **Array** poate avea deci una sau mai multe dimensiuni și pe fiecare dimensiune câte un număr de până la  $2^{31}-1$  elemente (limita permisă de memorie). Elementele componente ale unui **array** pot fi numerice, booleene, string-uri, valori ale unei forme de undă (grafic) etc, cu restricția de a fi toate de același tip. **Array**-urile sunt utile în cazul operării cu colecții de date similare și când se efectuează calcule repetitive.

Pentru a fi dispus în panou, un **Array** se selectează din meniul **Array & Cluster** al paletii **Controls**. Atunci când **Array**-ul se dispune pe panou, tipul datelor pe care acesta le va conține nu este încă definit. Se spune că elementul de tip **Array** este vid. Terminalul său din diagramă este de culoare neagră. Săgeata **Run** este spartă, semn că programul este incomplet și nu poate fi rulat.

Pentru a defini tipul de date, în elementul de tip **Array** trebuie introdus un control sau indicator scalar de tipul dorit (numeric, boolean sau alfanumeric). Elementul scalar ce urmează a fi introdus în **Array** poate exista deja pe panou sau poate fi selectat la momentul respectiv din paleta **Controls**.



Fig.3.3. Crearea unui tablou de scalari

Când scalarul este adus deasupra elementului **Array**, înainte de a se elibera butonul stâng al mouse-ului, conturul **Array**-ului este marcat cu o linie întreruptă ca în fig.3.3. **Array**-ul sesizează că i se va introduce un element scalar.

La eliberarea butonului mouse-ului, scalarul apare în interiorul elementului **Array**, acesta din urmă modificându-și corespunzător dimensiunile. În diagramă terminalul scalarului dispare, iar terminalul **Array**-ului capătă culoarea, textul și grosimea de chenar corespunzătoare datelor pe care le va manipula.

În partea sa stângă, un element de tip **Array** dispune de un index care indică numărul de ordine al valorii afișate în poziția superioară (indicele valorii). Indicele primei valori dintr-un **Array** este 0.

Imediat după ce i se definește tipul de date, un **Array** nu conține nici o valoare (este vid). Faptul este semnalizat prin afișarea "opacă" a elementelor pe care **Array**-ul le conține.

În mod implicit, atunci când este creat, un **Array** are o singură dimensiune, putând fi descris drept o matrice coloană. Numărul de dimensiuni ale unui **Array** poate fi modificat prin dimensionarea indexului acestuia. Se poziționează cursorul

mouse-ului pe un colț al indexului, astfel încât pe colțurile acestuia să apară patru simboluri "echer", se apasă butonul mouse-ului și se deplasează mouse-ul ținând butonul apăsat, până când indexul capătă dimensiunea dorită. Dacă un **Array** are mai mult de o dimensiune, se poate stabili afișarea mai multor linii și mai multor coloane, ca în fig.3.4.

În diagrama bloc, terminalul unui element **Array** se caracterizează prin prezența unor paranteze drepte, a căror grosime este proporțională cu numărul de dimensiuni. Atunci când se realizează legături în diagramă, conexiunile pe care se transmite un **Array** cu o dimensiune sunt mai groase decât cele destinate valorilor scalare. Pentru **Array**-urile cu două sau mai multe dimensiuni, conexiunile sunt reprezentate cu linie dublă. Spațiul intermediar al liniei duble este cu atât mai mare cu cât numărul de dimensiuni ale **Array**-ului este mai mare, după cum se exemplifică în fig.3.5.

Pentru a obține în mod dinamic (programatic) în diagrama bloc structuri de

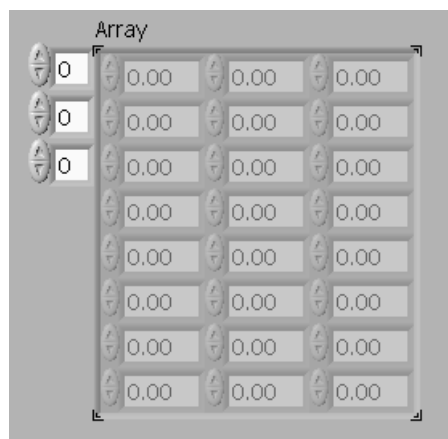


Fig.3.4. Redimensionarea **Array**-ului

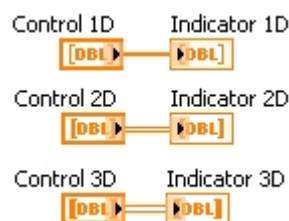


Fig.3.5. Terminalele **Array**-urilor de diferite dimensiuni

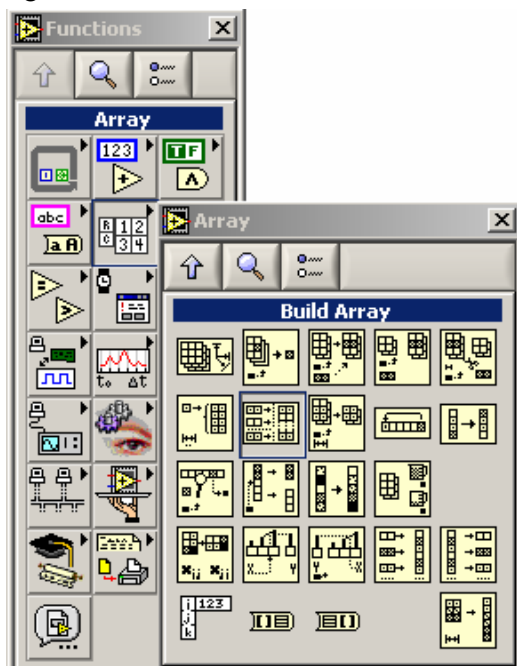


Fig.3.6. Subpaleta **Array** din paleta **Functions**

valori de tipul **Array**, pot fi utilizate o serie de funcții specifice. Aceste funcții, similar ca pentru **String**-uri, se găsesc în subpaleta **Array** din componența paletelor **Functions**. În fig.3.6 este redată subpaleta **Array** în care a fost selectată funcția **Build Array**. Această funcție permite introducerea în diagrama bloc a unui **Array** ce permite concatenarea mai multor **array**-uri, sau adaugă elemente la un **array**. Procedura de selectare și de vizualizare a funcțiilor simbolizate de icoanele din subpaleta **Array** este aceeași cu cea descrisă pentru **String**-uri.

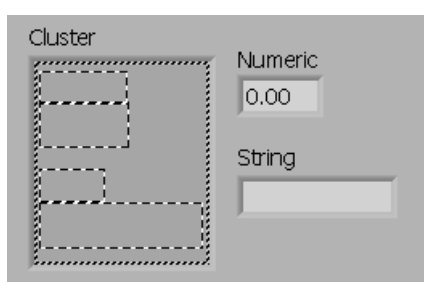


Fig.3.7. Definirea unui **Cluster** (așa cum se poate observa, un **cluster** poate fi văzut ca o bază de date simplă)

### Cluster-e

Elementele de tip **Cluster** sunt structuri de date ce pot conține mai multe valori, de tipuri și dimensiuni diferite. Un **Cluster** poate conține, de exemplu, o valoare numerică, două booleene și una alfanumerică (text), sau poate conține o combinație de valori scalare, vectoriale și chiar alte elemente de tip **Cluster** (de exemplu, o valoare numerică, un **Array** de valori booleene și un **Cluster** format din trei valori alfanumerice și una booleană).

Pentru a fi introdus în panoul unei aplicații, un element de tip **Cluster** se selectează din meniul **Array & Cluster** al paletelor de funcții. Imediat după ce este dispus pe panou, un **Cluster** este o structură vidă (un chenar), nu conține nici o valoare (fig.3.7).

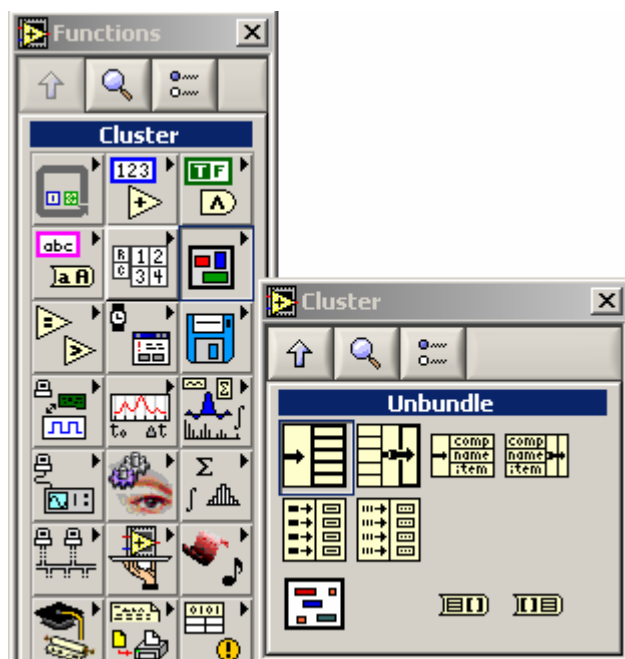


Fig.3.8. Subpaleta **Cluster** a paletelor **Functions**

Definirea structurii de valori dintr-un **Cluster** se realizează prin introducerea în chenarul său a unor elemente de tipurile dorite. Similar modului de definire a tipului elementelor **Array**, atunci când un element este poziționat deasupra chenarului unui **Cluster**, înainte ca utilizatorul să ridice butonul stâng al mouse-ului, chenarul elementului respectiv este reprezentat cu linie întreruptă.

Funcțiile specifice elementelor de tipul **Cluster** sunt grupate în meniul cu același nume al paletelor de **Functions**, reprezentată în fig.3.8

În cazul reprezentat pe fig.3.8 a fost selectată icoana corespunzătoare funcției denumită **Unbundle** care permite separarea elementelor componente ale unui **Cluster**. Pentru vizualizarea funcțiilor și detaliilor corespunzătoare diverselor icoane incluse în această subpaletă se procedează la fel ca în cazurile anterioare, descrise pentru **String-uri** și **Array-uri**.

### 3.2. Structuri de programare în LabVIEW

Structurile de programare sunt componente ale diagramei ale căror simboluri se găsesc în subpaleta **Structures** a paletelor de funcții având formele din fig.3.9. Fiecare structură are un contur distinct în interiorul căruia se poate include o parte din diagrama bloc (este denumită subdiagramă) și care funcționează conform cu regulile respectivei structuri. În diagramă, o structură de programare urmează să conțină o serie de terminale de elemente, simboluri de funcții și valori constante ce vor forma un flux de date în interiorul structurii. Dacă, în momentul în care structura este aleasă din subpaleta **Structures**, o parte din elementele pe care urmează să le conțină se află deja dispuse în diagramă, chenarul structurii poate fi trasat în jurul componentelor respective. După selectarea din subpaletă, se apasă butonul stâng al mouse-ului în unul din colțurile viitorului chenar și se deplasează mouse-ul ținând butonul apăsat. Pe măsură ce mouse-ul este deplasat, se trasează automat un contur cu linie întreruptă, contur ce va fi înlocuit de către structura selectată în momentul eliberării butonului mouse-ului.

După dispunerea în diagramă, conturul structurii poate fi deplasat sau dimensionat. Atunci când o structură este deplasată, toate elementele din interiorul său se deplasează odată cu ea. Elemente din afara structurii pot fi mutate în interiorul

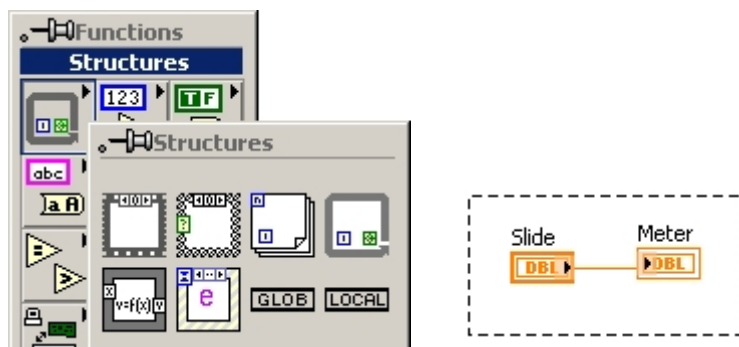


Fig.3.9. Simbolurile structurilor și încadrarea elementelor

acesteia, iar elemente din interior pot fi mutate în afara structurii. Dacă, prin dimensionare sau deplasare, o structură ajunge să se suprapună cu alte elemente din diagramă, acestea nu vor fi incluse în structură.

În cele ce urmează se vor prezenta detalii cu privire la structurile repetitive (de tip buclă) **For Loop** și **While Loop**, la structura de tip cauzal **Case** și la structura de tip secvențial **Sequence**.

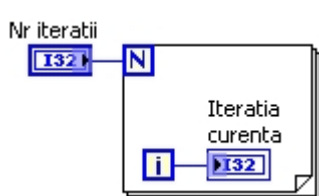


Fig.3.10. Bucla **For**

### Bucla For

O buclă **For** este o structură repetitivă care execută de un număr fixat de ori porțiunea de diagramă pe care o conține (subdiagramă). O execuție a porțiunii interne de diagramă poartă numele de iterație. O buclă **For**, redată în fig.3.10, dispune în colțul stânga-sus de un terminal (notat cu N), la care trebuie legată o valoare care să specifice numărul de iterații pe care bucla urmează să

le execute. În interiorul buclei se află un **terminal numărător** (notat cu i), care generează o valoare naturală reprezentând **indicele iterației curente** (aflate în curs de execuție). Indicii iterațiilor sunt numerotați începând cu valoarea 0, astfel încât ultima iterație executată de către bucla **For** va avea indicele N-1.

### Bucla While

Bucla **While** este o structură repetitivă cu condiție de terminare, adică execută porțiunea de diagramă pe care o conține până când la terminalul de continuare (aflat în colțul dreapta-jos) apare o anumită valoare booleană. Atunci când o buclă **While** este dispusă în diagramă, terminalul său de continuare se află în starea implicită **Continue If True**. Bucla va continua să execute iterații succesive atât timp cât la terminalul său de continuare ajunge o valoare logică **True**. Dacă la terminalul de continuare ajunge o valoare logică **False**, bucla **While** nu mai trece la următoarea iterație ci își încheie execuția. Starea implicită a terminalului de continuare poate fi modificată. Terminalul poate fi trecut în starea **Stop If True**, fie cu unealta de operare, fie apăsând butonul drept al mouse-ului și selectând starea din meniul propriu. În starea **Stop If True** bucla **While** își va încheia execuția atunci când la terminalul de continuare ajunge valoarea logică **True**. În colțul stânga-jos, bucla **While** dispune de un **terminal**

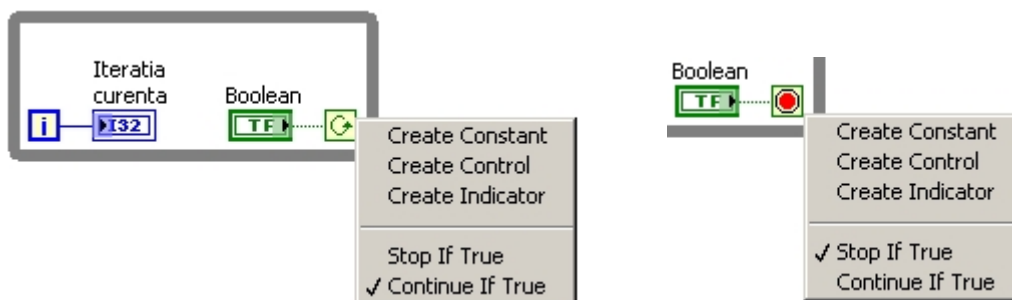


Fig.3.11. Bucla **While**

**numarator** (notat cu **i**), care generează o valoare naturală reprezentând **indicele iteratiei curente** (aflate în curs de execuție). Indicii iterațiilor sunt numerotați începând cu valoarea 0. Caracteristicile buclei **While** se pot observa în fig.3.11.

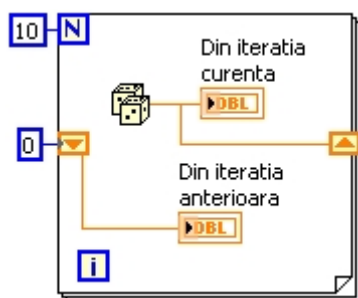


Fig.3.12. Registru de deplasare dispus pe conturul buclei

Structurile repetitive pot utiliza pentru transferul unor valori între două sau mai multe iterații successive **registri de transfer**. Pentru a adăuga un registru de transfer unei structuri repetitive (bucloa **For** sau **While**), se selectează opțiunea **Add Shift Register** din meniul propriu al structurii. Un registru de transfer conține două **terminale** (stâng și drept) dispuse pe conturul structurii ca în fig.3.12. O valoare conectată la terminalul drept al registrului va "apare" în terminalul stâng la iterația următoare.

O structură repetitivă poate conține mai mulți registre de transfer, fiecare cu una sau mai multe componente ale terminalului stâng și fiecare transferând între iterații valori de diverse tipuri.

### Structura Case

Atunci când este dispusă în diagramă, o structură **Case**, redată în fig.3.13, conține două ferestre (**True** și **False**), fiecare dintre acestea cu propriul flux de date.

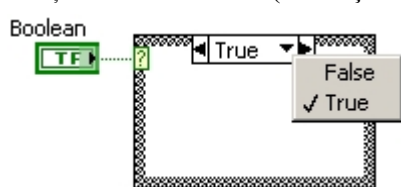


Fig.3.13. Structura Case

Pe conturul structurii se află dispus un **terminal selector** (de culoare verde, conținând semnul întrebării). Structura **Case** va executa doar una din cele două ferestre, în funcție de valoarea booleană (**True** sau **False**) conectată la terminalul său selector.

Dacă la terminalul selector al unei structuri **Case**, în locul unei valori booleene se conectează o valoare numerică (număr întreg sau natural), numele ferestrelor se vor modifica automat din **True** și **False** în **0** și **1**. Structura va executa fereastra al cărei nume va coincide cu valoarea numerică ce ajunge la terminalul său selector.

În această situație, are sens ca structura **Case** să conțină mai mult de două ferestre.

Deoarece structura **Case** nu va putea conține atâtea ferestre câte valori numerice pot ajunge la terminalul său selector, una dintre ferestre trebuie declarată drept **fereastră implicită (Default)**. Fereastra implicită va fi executată atunci când valorii ajunse la terminalul selector nu îi corespunde nici un nume de fereastră.

Un nume de fereastră poate conține și descrierea unei mulțimi de valori, de genul 2..5, 7..9, înțelegând prin aceasta că fereastra respectivă va fi executată dacă la terminalul selector ajunge una din valorile 2, 3, 4, 5, 7, 8 sau 9. O valoare nu poate exista în numele mai multor ferestre (ar exista o incertitudine privind fereastra ce trebuie executată).



Dacă la terminalul selector al unei structuri **Case** se conectează o valoare alfanumerică, atunci numele ferestrelor vor trebui să fie de tipul **String** (se modifică utilizând unealta de editare texte). Și în această situație trebuie să existe o fereastră implicită. Se pot, de asemenea, defini mulțimi de valori (de genul "A".."F"), cu condiția evitării dublurilor.

O valoare introdusă din exterior într-o structură **Case** este accesibilă în orice fereastră a acesteia. Dacă dintr-o fereastră a unei structuri **Case** se scoate o valoare în exterior, pe conturul structurii apare un **terminal de ieșire** de culoare albă, semn că diagrama conține o eroare. Atunci când o structură **Case** are pe contur un terminal de ieșire, la acesta trebuie definite legături *din toate ferestrele structurii*. În caz contrar, se spune că ieșirea din structură *nu este complet definită*. Dacă ieșirea este complet definită, terminalul de ieșire se umple cu culoarea reprezentând tipul de date conectate (pentru a fi corect definită, la o ieșire trebuie să ajungă *același tip de date* din toate ferestrele structurii. Aceste detalii sunt redată în fig.3.14.

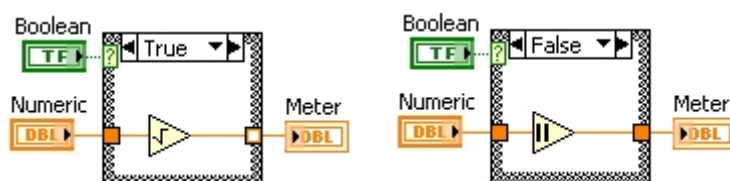


Fig.3.14. Structuri **Case** cu terminale pe contur

### Structura Sequence

O structură secvențială, de regulă dispune de mai multe ferestre (subdiagrame), fiecare dintre acestea cu propriul flux de date. Structura **Sequence** execută în ordine ferestrele respective. De exemplu, în fig.3.15 sunt reprezentate două ferestre ale unei structuri **Sequence**. Structura va executa întâi fereastra cu indicele **0**, trimițându-se valoarea de la elementul de control **Slide** la elementul indicator **Meter**, apoi va executa fereastra **1** ce are ca efect transmiterea valorii booleene între cele două elemente.

Atunci când este dispusă în diagramă, o structură **Sequence** conține o singură fereastră, aceasta fiind totodată și fereastra curentă (vizibilă).

Deschizând meniul propriu al structurii și alegând una din opțiunile **Add Frame After** sau **Add Frame Before**, se poate adăuga o fereastră după sau înaintea celei curente.

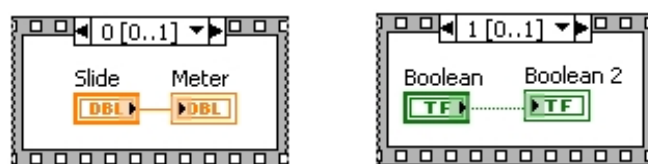


Fig.3.15. Două ferestre ale unei structuri **Sequence**

Dacă o structură **Sequence** conține mai mult de o fereastră, atunci ferestrele primesc **indici** (numere de ordine) începând cu **0**, iar pe latura superioară a conturului structurii apare un **selector** prin intermediul căruia se poate trece de la o fereastră la alta, ca în fig.3.16.

Trecerea la o altă fereastră se poate realiza, fie prin intermediul celor două săgeți de la capetele selectorului, fie apăsând pe săgeata din partea dreaptă (îndreptată în jos) și selectând indicele ferestrei dorite din lista care se desfășoară.

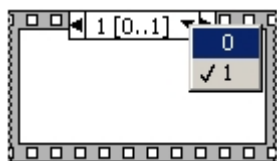


Fig.3.16. Indexarea ferestrelor structurii **Sequence**

La un moment dat, este vizibil doar fluxul din fereastra curentă a unei structurii **Sequence**. Se poate crea o dublură a ferestrei curente, selectând din meniul propriu al structurii opțiunea **Duplicate Frame**. Fereastra curentă poate fi ștearsă, cu opțiunea **Delete This Frame**. Atunci când o structură **Sequence** conține cel puțin două ferestre, ordinea acestora poate fi schimbată astfel: se modifică indicele ferestrei curente, selectând

opțiunea **Make Frame**, apoi noul indice. Atunci când se dorește transferul unei valori între două ferestre ale unei structurii **Sequence**, prin selectarea opțiunii **Add Sequence Local** se dispune pe conturul structurii o variabilă locală a acesteia (sub forma unui pătrat galben). În meniul propriu există și alte opțiuni privind legăturile cu fluxul de date externe.

### 3.3. Calculul mediei și dispersiei - metoda I

În cadrul acestui exemplu se va efectua calculul mediei și dispersiei unei colecții variabile de numere aleatoare. Programul se poate considera unul de test, deci se va putea executa în modul **Run Continuously**.

În panoul frontal al aplicației se adaugă un control digital care va indica numărul de numere aleatoare generat și i se stabilește valoarea implicită 10. Tipul de date pe care îl va genera trebuie să fie întreg, de aceea se va alege **Representation >> Unsigned Byte**. Pentru a afișa valorile se vor crea două tablouri de întregi. Din grupul **Array & Cluster** se selectează **Array** și se plasează primul tablou în cadrul panoului. Apoi, din subpaleta **Numeric** se alege un indicator digital, se dispune inițial pe panou, i se atribuie precizia la 3 zecimale și apoi se trage în interiorul tabloului. În acest moment s-a creat un tablou de numere întregi. Se redimensionează tabloul pe orizontală astfel încât să rezulte 10 elemente vizibile. În dreptul indexului se apasă butonul dreapta al mouse-ului și din meniul **Visible Items** se execută click pe **Index Display** pentru a ascunde index-ul tabloului și se modifică eticheta tabloului în **Sir cu N nr aleatoare**. S-a setat mai sus în așa fel încât la prima rulare a programului să se genereze 10 numere aleatoare. Dacă utilizatorul selectează mai multe, nu se vor afișa decât 10 în tablou. Pe de altă parte, dacă se alege un număr mai mic, în celulele rămase se va afișa valoarea 0 și ele vor fi colorate în gri pentru a semnaliza că sunt nefolosite.

Se selectează tabloul nou creat și cu **Copy/Paste** este creat cel de-al doilea tablou, care se va intitula **Sir ce are ultima valoare Suma**. Acest tablou se redimensionează pentru a conține 11 elemente, în care ultimul va fi suma celorlalte. Pentru a marca acest aspect, folosind butonul **Edit Text** din paleta **Tools** se efectuează click în zona ultimului element al noului tablou și se scrie **Suma**. Astfel a fost creat un tabel nou care indică faptul că în ultimul element al tabloului se găsește suma (fig.3.17).



Fig.3.17. Panoul frontal inițial al aplicației **Calculul mediei și dispersiei metoda I**

Pentru evidențiere, suma se va afișa și în cadrul unui indicator digital separat de cele două tablouri. Scopul programului este de a calcula media și dispersia celor  $N$  numere generate aleator. Se crează încă două indicatoare pentru cele două valori și se personalizează pentru a arăta ca în fig.3.18. Pentru aceasta se selectează, din spațiul butonului de colorare din paleta **Tools**, butonul din fundal și se alege culoarea dorită, de exemplu roșu. După aceea se stabilește precizia indicatoarelor la 4 zecimale.

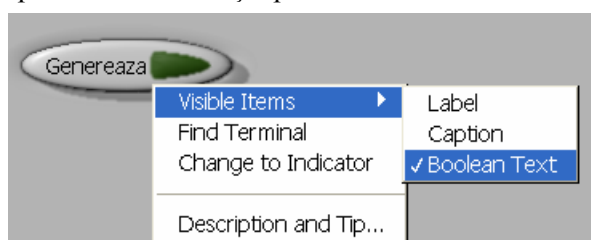


Fig.3.18. Activarea **Boolean Text** în cazul unui buton

După cum se poate observa în fig.3.18, a fost adăugat un buton tip **Push Button** din grupul **Boolean**, căruia i s-a ascuns eticheta și a fost redenumit **Genereaza**. Trebuie menționat că, dacă deasupra unui buton se găsește eticheta sa sau **caption**-ul, atunci în acea zonă butonul nu reacționează. Chiar dacă se încearcă o ordonare, în sensul să se “trimită” eticheta în spate cu opțiunea **Move to Back**, nu se rezolvă nimic, pentru că butonul nu este transparent și eticheta va fi ascunsă. Pentru ca

totuși să apară text deasupra butonului și acesta să reacționeze chiar dacă se face click în zona textului, se va activa din **Visible Items** opțiunea **Boolean Text** și se va redenumi **Genereaza** ca în fig.3.18. În acest mod, butonul va reacționa indiferent în ce zonă a sa a primit un click. Scopul acestui buton este de a produce o iterație atunci când este apăsat. Pentru aceasta trebuie să i se modifice modul de funcționare. În mod implicit, un buton funcționează ca un switch de tip boolean: când nu este apăsat se generează valoarea **False**, iar când este apăsat se generează valoarea **True**. În cazul de față, butonul trebuie să genereze **True** doar când este apăsat și apoi să revină și să genereze în continuare **False**. Valoarea **True** trebuie să fie generată doar o singură dată, pentru ca utilizatorul să poată citi datele. Comportamentul butonului la acțiunea utilizatorului se poate stabili din meniul care apare în urma apăsării butonului dreapta al mouse-ului în zona acestuia și care este redat în fig.3.19. Din acest meniu se alege grupul **Mechanical Action** și apoi **Latch When Pressed**. Acum butonul va genera o singură valoare **True** când va fi apăsat și apoi va reveni automat la starea inițială și anume **False**. După cum se observă, comportamentul butoanelor poate fi modificat în mai multe feluri, în funcție de nevoile utilizatorului. Descrierea comportamentului pentru fiecare acțiune se poate vedea consultând **Help-ul LabVIEW-ului** și tastând în câmpul **Search** opțiunea **Change mechanical action**. Primul item găsit este o descriere amănunțită a fiecărui tip de acțiune a unui buton boolean, așa cum se arată în fig.3.19.

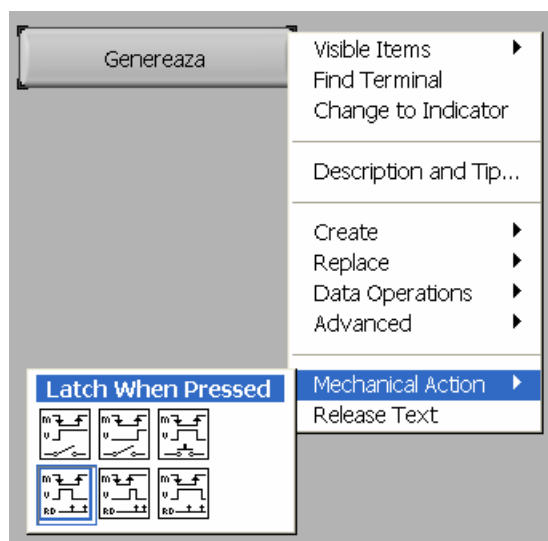


Fig.3.19. Selectarea tipului de date pentru controlul timpului de reafixare a rezultatelor

Cu această operație s-a terminat lucrul asupra panoului frontal și se trece la partea de diagramă bloc.

Diagrama bloc finală a aplicației va fi destul de complicată, de aceea se va descrie pas cu pas realizarea ei.

Controlul digital pentru numărul de valori generate se poziționează în partea stângă a diagramei. Din grupul **Structures** al paletei **Functions** se alege **For Loop**, se

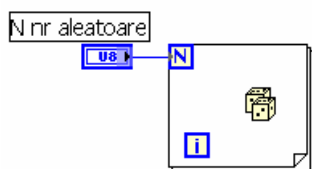


Fig.3.20. Bucla **For** pentru generatorul de numere aleatoare

plasează în vecinătatea terminalului controlului digital și se conectează acesta din urmă cu **N**. În interiorul buclei se adaugă un generator **Random Number (0-1)** care se selectează din grupul **Numeric**. Bucla **For** va arata ca în fig.3.20.

Se mută **Array**-ul pentru afișarea numerelor aleatoare lângă **For Loop** și se conectează cu generatorul de numere aleatoare. În acest moment, s-ar putea ca firul care conectează bucla **For** cu **Array**-ul să fie întrerupt. Acest lucru se datorează faptului că valorile din buclă nu sunt indexate ca să poată fi adăugate în tablou. Pentru a realiza indexarea se efectuează click dreapta pe conectorul de pe buclă și se activează opțiunea de indexare, după cum se poate observa pe fig.3.21.

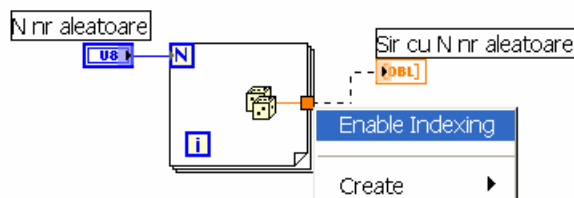


Fig.3.21. Activarea indexării elementelor generate pentru a putea fi adăugate în tablou

În continuare, urmează să se realizeze suma valorilor din **Array**, care se vor afișa, alături de restul numerelor, în al doilea tablou. Pentru aceasta se folosește modulul **Add Array Elements** din grupul **Numeric**. Ieșirea acestuia se conectează la indicatorul digital pentru **Suma**, care a fost prevăzut în panoul frontal. Pentru a adăuga elementul sumă la **Array**-ul deja existent, va trebui să se folosească un modul tip **Build Array**, care se selectează din grupul **Array** al paletii **Functions** și a cărui utilizare este ilustrată în fig.3.22.

Când se adaugă modulul în diagramă, se observă că acesta are o singură intrare și o ieșire. Se selectează modulul și apoi se trage de partea sa inferioară. Se constată că numărul de intrări crește. Intenționând să se adauge un element într-un

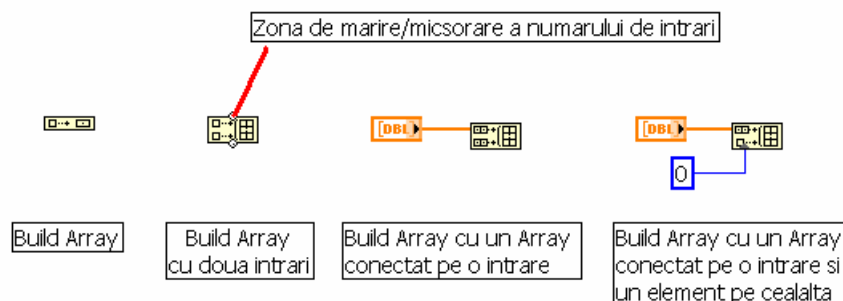


Fig.3.22. Modificarea automată a numărului de intrări ale unui **Array** și adăugarea diferitelor tipuri de elemente la intrările acestuia

**Array** deja existent va fi nevoie doar de două intrări. Un alt aspect care trebuie remarcat este că icoana modulului creat se schimbă în momentul când pe o intrare se adaugă elemente tip **Array**.

Ieșirea de la modulul **Build Array** se conectează la cel de-al doilea **Array**, rezultând o schemă precum cu cea din fig.3.23.

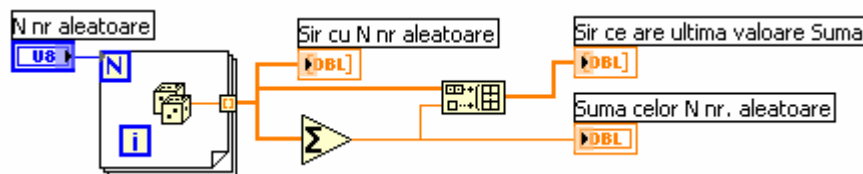


Fig.3.23. Calculul sumei elementelor din tablou

În continuare, se calculează media și dispersia elementelor tabloului folosind relațiile matematice cunoscute: media este  $M = \frac{1}{n} \sum_{i=1}^n x_i$ , iar dispersia este

$D = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - M)^2}$ . Pentru fiecare valoare care se dorește să se obțină, trebuie să se construiască diagrama bloc, care să fie conformă ecuației matematice atașate.

Pentru medie este nevoie de un modul de sumare a elementelor tabloului **Add Array Elements**, care se selectează din paleta **Functions**, meniul **Numeric**. Numărul  $n$  de elemente ale tabloului se determină cu funcția **Array Size** în meniul **Array** din paleta **Functions**.

Atât modulul de sumare a elementelor **Array**-ului, cât și **Array Size** generează valori scalare, care se vor împărți și se va obține media  $M$  care este tot un scalar. Diagrama parțială, cu ramura pentru medie, arată ca în fig.3.24.

Dispersia necesită cunoașterea valorii medii a elementelor deoarece, așa cum se vede din relația de calcul din fiecare element al tabloului trebuie scăzută valoarea medie.

Este de remarcat că în mediul **LabVIEW** datele de tip **Array** sunt percepute ca cele de tip scalar. De exemplu, pentru a transfera un **Array** în altul este de ajuns să se conecteze cele două tablouri printr-un simplu fir. Pentru a ridica la putere elementele unui **Array** se conectează ieșirea la ambele intrări ale unui modul de înmulțire, iar rezultatul va fi un **Array** similar, care are elementele la pătrat. O altă

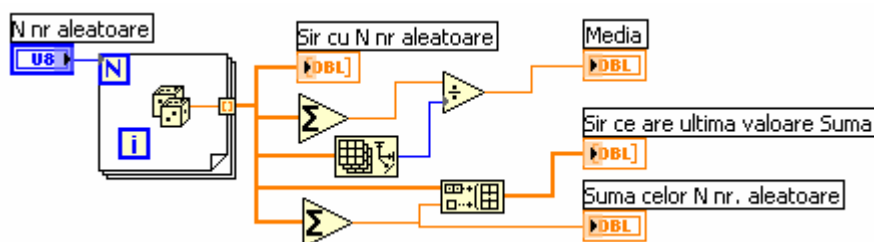


Fig.3.24. Calculul sumei și a mediei elementelor dintr-un **Array**

situație este cea în care se dorește să se adune la fiecare element al unui tablou câte un scalar. Pentru aceasta, este suficient să se conecteze la o intrare a modulului sumă tabloul și la cealaltă scalarul. Operația este similară pentru toate operațiile matematice de bază: scădere, înmulțire, împărțire.

Conform celor de mai sus, se adaugă un modul de scădere la ale cărui intrări se conectează **Array**-ul inițial și media calculată. În continuare, trebuie să se ridice la pătrat noile elemente ale **Array**-ului și acest lucru se realizează cu ajutorul unui modul de înmulțire. Se calculează suma elementelor obținute, care apoi se împarte la numărul de elemente minus 1, pentru a respecta formula.

Se obține diagrama bloc din fig.3.25, care permite calculul sumei elementelor, al mediei și al dispersiei.

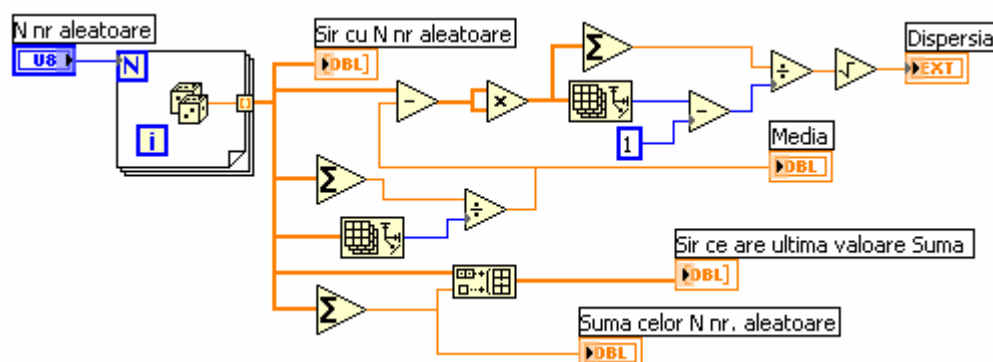


Fig.3.25. Diagrama bloc pentru **Calculul mediei și dispersiei - metoda I**

După ce s-a realizat schema pentru calculul mediei și dispersiei, trebuie să se

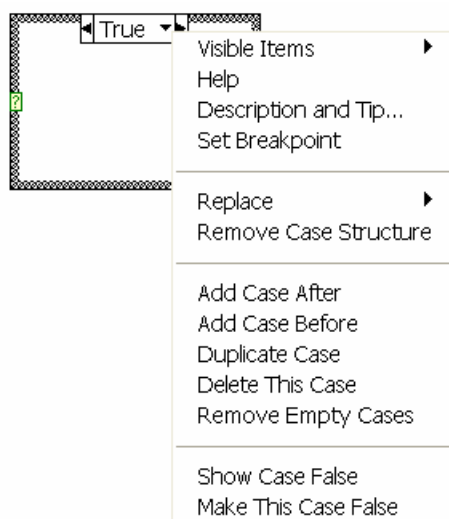


Fig.3.26. Structura **Case** și meniul de modificare a acesteia, obținut prin click dreapta în zona etichetei **True**

adauge structurile necesare pentru generarea numerelor aleatoare la apăsarea butonului introdus în panoul frontal. Pentru aceasta este necesară o sintaxă de tip **If-else**, la fel ca într-un program obișnuit. **LabVIEW**-ul nu are prevăzut explicit o structură de acest tip, însă se poate implementa orice situație tip **If-else** cu ajutorul structurii **Case**.

Structura **Case** are în mod implicit, două cazuri: **True** și **False**. Însă acestea se pot redenumi în funcție de ce condiție dorim să fie îndeplinită. De asemenea, se pot adăuga alte situații unei structuri **Case**.

După cum se observă în fig.3.26, bucla **Case** are o intrare reprezentată de o căsuță cu simbolul "?". Această căsuță este zona de conectare a buclei cu exteriorul. În funcție de valoarea primită în această căsuță, se execută una din ramurile buclei **Case**.

După cum s-a mai menționat, structura **Case** are în mod implicit două ramuri: **True** și **False**. Se observă că în zona etichetei **True** se află două săgeți laterale și una orientată în jos. Indiferent de unealta cu care se lucrează, adică indiferent de cursorul selectat, atunci când se ajunge în zona etichetei cursorul se modifică într-o săgeată cu direcția de deplasare (stânga sau dreapta). Cu săgețile laterale se pot executa secvențial deplasări între ramurile **Case**-ului, în timp ce prin apăsarea săgeții orientată în jos, se deschide un meniu tip pop-up, redat în fig.3.27, de unde se poate selecta ramura **Case**-ului care interesează.

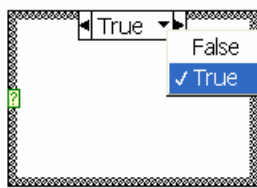


Fig.3.27. Selectarea unei ramuri în cadrul unei structuri **Case**

O structură **Case** se poate adăuga similar uneia **For** sau **While**. Când se selectează din **Functions** subpaleta **Structures** și apoi **Case Structure**, cursorul se transformă într-un dreptunghi. Se alege colțul din stânga sus și apoi se încadrează schema, sau porțiunea dorită din schemă, în interiorul **Case**-ului.

În cazul de față va trebui să se includă toată schema dezvoltată pentru calculul mediei și dispersiei în cadrul buclei **Case**, cu excepția butonului de comandă a generării numerelor aleatoare. Când butonul este apăsător, se generează o valoare **True**. Deci toată diagrama va rămâne pe ramura **True** a **Case**-ului. Pe **False** nu interesează să se întâmple un eveniment, deci se va lăsa goală. În continuare se leagă butonul **Genereaza** la intrarea buclei **Case**. Diagrama finală a aplicației va arăta ca în fig.3.28.

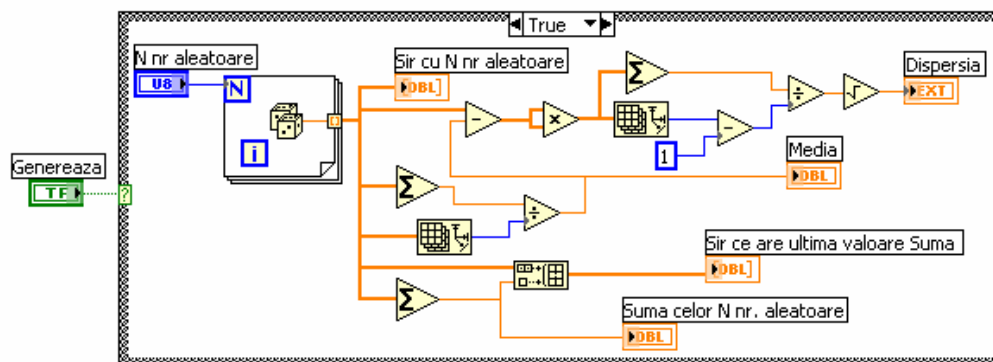


Fig.3.28. Diagrama finală a aplicației

Aplicația este funcțională și trebuie să se revină în panoul frontal pentru a o executa. Așa cum s-a stabilit la început, aplicația va rula în modul **Run Continuously** și va fi oprită cu **Abort Execution**. După cum se poate observa, se generează un număr inițial de 10 numere aleatoare, care sunt afișate și cărora li se calculează media și dispersia. Numerele aleatoare nu sunt generate încontinuu, ci doar la apăsarea butonului plasat în panoul frontal. Salvarea aplicației se face sub denumirea **Calculul mediei și dispersiei metoda I\_[nume / prenume student]\_[grupa].vi**.



Programul are un bug la afișare. Atunci când colecția de numere ce urmează a fi generate este mai mică decât 10, suma este afișată în următoarea celulă goală a **Array**-ului și nu în ultima, așa cum indică eticheta. Panoul frontal va arăta, în acest caz, așa cum este prezentat în fig.3.29.

Fig.3.29.Panoul frontal final al aplicației **Calculul mediei și dispersiei metoda I**

### 3.4. Calculul mediei și dispersiei – metoda II

Următoarea aplicație calculează media și dispersia unor numere aleatoare ce vor avea o bază selectată de utilizator prin intermediul unui **Slide**. Numerele generate vor fi în intervalul [bază, bază+1]. Valorile se vor genera continuu, până când aplicația va fi oprită. Intervalul de generare este prestabilit la 200 ms.

Primul pas va fi construcția panoului frontal. **Slide**-ul ce reprezintă baza numerelor aleatoare se copiază din **Calcul volum cilindru** [nume / prenume student]\_[grupa].vi cuprinsă în **Aplicația II.2**, pentru a nu repeta pașii care au fost efectuați acolo. Se deschide aplicația anterior precizată și, cu ajutorul unelei

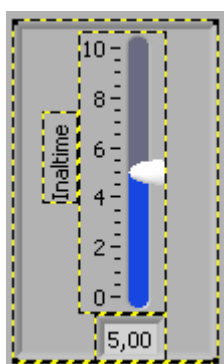


Fig.3.30. Grupul **Înălțime** din aplicația **Calcul volum cilindru**

**Select/Size**, se încadrează **slide-ul Înălțime**, împreună cu indicatorul digital și cu chenarul de decor. Se reamintește că aceste obiecte erau grupate, ca atare selecția va arăta ca în fig.3.30.

Prin procedeul **Copy/Paste** se aduce grupul **Înălțime** în noua aplicație, care se va numi **Calculul mediei și dispersiei metoda II**. Cu unealta **Edit Text**, se modifică eticheta **Înălțime** în **Baza**. Se stabilește valoarea implicită ca fiind 4, în loc de 5.

Se repetă procedura pentru **Waveform Chart** iar aici se ascunde eticheta **Înălțime**. Pentru aceasta, cu click dreapta în zona graficului, se alege **Y Scale** și apoi se deselectează opțiunea **Visible Scale Label** procedând ca în fig.3.31.

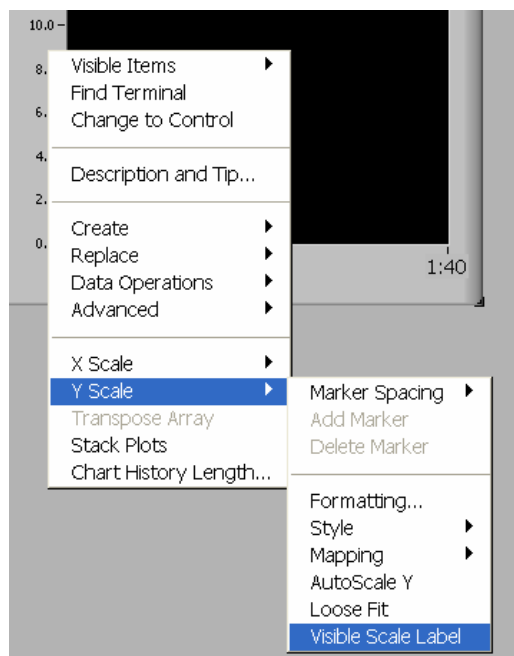


Fig.3.31. Ascunderea etichetei “Înălțime”

Valoarea implicită a indicatorului digital se stabilește la 4 pentru a corespunde slide-ului **Baza**.

Din exemplul anterior, **Calculul mediei și dispersiei metoda I**, având panoul frontal inițial redat în fig.3.17, se selectează cele două indicatoare pentru medie și dispersie “trăgând” un dreptunghi în jurul lor. Pentru a selecta și indicatorul pentru sumă, cu tasta **Shift** apăsată, se efectuează click în zona indicatorului **Suma celor N nr. aleatoare**. Se observă că în acest moment sunt selectate cele trei indicatoare, care se aduc prin **Copy/Paste** în noua aplicație.

Se redenumbește indicatorul **Suma celor N nr. aleatoare** în **Suma valorilor**. Apoi, se mai adaugă un indicator în care se vor reține atâtea numere aleatoare câte au



Fig.3.32 Panoul frontal al aplicației

fost procesate în cadrul aplicației. Pentru a opri aplicația, este necesar un buton tip **Stop**.

După ce au fost adăugate elementele sus-menționate, ele se aranjează într-un mod convenabil așa cum se poate observa în fig.3.32.

Pasul următor este reprezentat de realizarea diagramei bloc a aplicației. Se are în vedere generarea unei serii de numere aleatoare, la un interval de 200 ms. Utilizatorul va putea interveni asupra **Slide**-ului, selectând o nouă bază pentru numerele generate. Când se va apăsa butonul **Stop** se vor calcula media și dispersia numerelor și se vor afișa pe ecran, după care aplicația se va opri.

S-au identificat cele două etape ale aplicației. În prima etapă se va trata partea de generare. Pentru aceasta, se conectează **Slide**-ul **Baza** cu indicatorul aferent și se adaugă un modul de adunare. La una din intrările acestuia va fi un fir de la **Baza**, iar cel de-al doilea va fi conectat la un generator de numere aleatoare, după ce, în prealabil, a fost și acesta introdus în schemă. Ieșirea modulului va fi conectată atât la indicatorul digital pentru valoarea instantanee, cât și la indicatorul grafic.

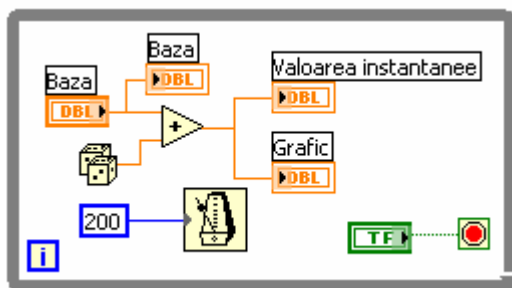


Fig.3.33. Generarea numerelor aleatoare

Se adaugă partea de temporizare, realizată dintr-o constantă cu valoarea 200 și un **Wait Until Next ms Multiple** și se aduce butonul **Stop** în vecinătatea schemei.

Se încadrează elementele într-o buclă **While**, care se va opri atunci când va fi apăsat butonul **Stop**. S-a obținut astfel prima parte a diagramei și anume partea de generare a numelor aleatoare redată în fig.3.33.

După acest pas, pentru a putea calcula media și dispersia, trebuie să se stocheze valorile acestora într-un **array**. Se inserează un modul **Build Array**, a cărui intrare se conectează la firul care iese din modulul de sumare. Se observă că trecerea din interiorul spre exteriorul buclei **While** se face printr-un “tunel”. Se execută click dreapta asupra acestuia și se activează opțiunea **Enable Indexing**, pentru a putea stoca valorile în tablou.

Spre deosebire de exemplul anterior, nu se mai concatenează intrările tabloului în care vor fi stocate valorile. Pentru aceasta, trebuie să se verifice dacă opțiunea **Concatenate Inputs** este activată în meniul pop-up care se deschide la acționarea click dreapta în zona modulului **Build Array**. Dacă este activă trebuie dezactivată.

Care este diferența dintre un tablou cu intrările concatenate și unul fără? În cazul de față nu este o diferență semnificativă, dar în anumite cazuri se dovedește a fi folositoare. Pentru exemplificare, se consideră două tablouri {1,2} și {3,4,5}. Dacă se crează un tablou folosind modulul **Build Array** și opțiunea **Concatenate Inputs** este selectată, atunci rezultatul final va fi {1,2,3,4,5}. Dacă opțiunea mai sus-menționată nu este activă, atunci rezultatul va fi crearea unui tablou 2D: {{1,2,0},{3,4,5}}, în care a fost adăugată valoarea 0 pentru a se respecta dimensiunea.



lucrează cu o matrice cu 2 linii și n coloane. În acest sens, pentru a calcula numărul de elemente care este necesar pentru calculul mediei și dispersiei, va trebui să se afle numărul de coloane. **Array Size** va genera un vector cu cele două dimensiuni, de aceea va trebui să se aplice un **Index Array** pentru a stabili numărul de coloane. În momentul când se aplică **Index Array** și se extrage elementul al doilea (numărul de coloane), firul întrerupt va deveni continuu și astfel aplicația se va putea rula. Se reamintește că **Array**-urile în **LabVIEW** au primul element pe poziția 0, al doilea pe poziția 1 ș.a.m.d. Artificiul care trebuie realizat se observă în fig.3.35. Același procedeu se folosește și în cazul dispersiei, astfel că diagrama din fig.3.35 devine varianta finală a aplicației de **Calcul al mediei și dispersiei metoda II**, iar panoul frontal este cel din fig.3.32.

### 3.5. Chestiuni de studiat

1. Pentru fiecare dintre funcțiile prezentate pentru modul de grupare a datelor **String**, se va crea câte un program care să exemplifice modul de utilizare a funcției respective.

2. Se va insera în diagrama bloc o constantă **Array** cu valori numerice, având o singură dimensiune, conținând elementele 1, 2, 3, 4, 5. Apoi se va inițializa un **Array**, tot cu o dimensiune, care să cuprindă de zece ori 7. Se va construi o matrice având pe prima linie rezultatul concatenării celor două **Array**-uri precedente, iar pe linia a doua rezultatul intersecției acestora.

3. Se va dispune pe panoul frontal un element de tip **Cluster** și se va introduce în chenarul acestuia un element pentru valori numerice și un **Array** cu valori booleene. Similar, se va crea un alt **Cluster** care să conțină două elemente de tip **String**. Pentru cele două **Cluster**-e să se testeze în diagrama bloc funcțiile **Bundle** și **Unbundle**.

4. Se va realiza aplicația descrisă în secțiunea 3.3 **Calculul mediei și dispersiei metoda I**. Se vor crea panoul frontal și diagrama bloc în conformitate cu indicațiile cuprinse în această secțiune.

5. Se va proceda la fel ca la punctul 4 pentru realizarea aplicației descrisă în secțiunea 3.4 pentru calculul mediei și dispersiei prin cea de-a doua metodă.

### 3.6. Modul de lucru și prezentarea rezultatelor

Pentru punctul 1 de la **Chestiuni de studiat**, de exemplu, pentru funcția **String Length**, se va plasa în panoul frontal un element de control pentru valori alfanumerice, în care se va introduce un text și un indicator pentru valori numerice, programul trebuind să afișeze numărul de caractere al textului introdus. Prin proceduri similare se vor realiza programe pentru funcțiile **Concatenate Strings** și **String Subset**.

Referitor la punctele 2 și 3 se vor realiza programele cerute ținând seama de noțiunile cuprinse în secțiunea 3.1 și de detaliile pe care le oferă **Context Help** în ceea ce privește funcțiile din subpaletele **Array** și **Cluster**.

La punctul 4, se va rula aplicația realizată și se vor observa rezultatele obținute. Pe parcursul creării aplicației, se vor testa diversele proceduri utilizate pentru stabilirea caracteristicilor și proprietăților elementelor și structurilor introduse în panoul frontal și în diagrama bloc, ținând seama îndeosebi de noțiunile prezentate în secțiunile **3.2** și **3.3**.

Pentru punctul 5, ținând seama de aspectele de creare și de rulare a **VI**-ului, se va face o comparație de principiu între aplicația de calcul a mediei și dispersiei prin cea de a doua metodă, cu aplicația similară bazată pe prima metodă, luând în considerare, în principal, modul de generare a numerelor aleatoare și de afișare a rezultatelor. De asemenea, se vor compara panourile frontale și diagramele bloc ale celor două aplicații.

Aplicațiile realizate, observațiile și comentariile referitoare la rezultatele obținute se vor salva în fișierul personal al fiecărui student conform procedurii descrise la lucrările anterioare. O atenție deosebită se va acorda celor două aplicații de calcul a mediei și dispersiei și comparației dintre ele. Titlurile sub care se vor salva aplicațiile vor fi cele ale secțiunilor **3.3** și **3.4** din prezenta lucrare, sau similare cu acestea.